

# 小白成长日记——第八天

---

我变强了，也要秃了

## Node.js

### 基本模块

#### 1. global

在前面的JavaScript课程中，我们已经知道，JavaScript有且仅有一个全局对象，在浏览器中，叫window对象。而在Node.js环境中，也有唯一的全局对象，但不叫window，而叫global，这个对象的属性和方法也和浏览器环境的window不同。进入Node.js交互环境，可以直接输入：

```
> global.console
Console {
  log: [Function: bound ],
  info: [Function: bound ],
  warn: [Function: bound ],
  error: [Function: bound ],
  dir: [Function: bound ],
  time: [Function: bound ],
  timeEnd: [Function: bound ],
  trace: [Function: bound trace],
  assert: [Function: bound ],
  Console: [Function: Console] }
```

#### 2. process

process也是Node.js提供的一个对象，它代表当前Node.js进程。通过process对象可以拿到许多有用信息：

```
> process === global.process;
true
> process.version;
'v5.2.0'
> process.platform;
'darwin'
> process.arch;
'x64'
> process.cwd(); //返回当前工作目录
'/Users/michael'
> process.chdir('/private/tmp'); // 切换当前工作目录
undefined
> process.cwd();
'/private/tmp'
```

### 3. 同步与异步

同步操作的好处是代码简单，缺点是程序将等待IO操作，在等待时间内，无法响应其它任何事件。而异步读取不用等待IO操作，但代码较麻烦；

### 4. http-服务器获取本地文件

```
"use strict"
var
  fs = require("fs"),
  url = require("url"),
  path = require("path"),
  http = require("http");

var root = path.resolve(process.argv[2] || ".");
console.log("static root dir" + root);

var server = http.createServer(function(request,response){
  var pathName = url.parse(request.url).pathname;
  var filepath = path.join(root,pathName);
  fs.stat(filepath,function(err,stats){
    if(!err && stats.isFile()){
      console.log("200" + request.url);
      response.writeHead(200);
      fs.createReadStream(filepath).pipe(response);
    }else if(!err){
      var f = path.join(root,"index.html");
      fs.createReadStream(f).pipe(response);
    }else{
      console.log("404" + request.url);
      response.writeHead(404);
      response.end("404 Not Found");
    }
  });
});
server.listen(8080);
console.log("Server is running at http://localhost:8080/");
```